QTM151 PROJECT

MINSEO KIM, ELIZABETH PARK, ANDREW JEONG, SU KIM, DANIEL MIN

4/23/2019

bring data

```
library(readr)
data <- read_delim("C:/Users/mkim458/Desktop/data.txt", "\t", escape_double = FALSE, tri
m_ws = TRUE)
```

```
## Parsed with column specification:
## cols(
##   .default = col_double(),
##   date_time = col_datetime(format = ""),
##   site_name = col_character(),
##   user_location_country = col_character(),
##   user_location_region = col_character(),
##   user_location_city = col_character(),
##   user_location_latitude = col_character(),
##   user_location_longitude = col_character(),
##   orig_destination_distance = col_character(),
##   srch_ci = col_date(format = ""),
##   srch_co = col_date(format = ""),
##   hotel_country = col_character(),
##   distance_band = col_character(),
##   hist_price_band = col_character(),
##   popularity_band = col_character()
## )
```

```
## See spec(...) for full column specifications.
```

```
dest <- read_delim( "C:/Users/mkim458/Desktop/dest.txt", "\t", escape_double = FALSE, tr
im_ws = TRUE)
```

```
## Parsed with column specification:
## cols(
##   .default = col_double(),
##   srch_destination_name = col_character()
## )
```

```
## See spec(...) for full column specifications.
```

manipulate and random sample data for only US user

```
detach("package:readr", unload=TRUE)
library(tidyverse)
```

```
## -- Attaching packages ---------------------------------------------------------
----------------- tidyverse 1.2.1 --
```

```
## v ggplot2 3.1.1       v purrr   0.3.2
## v tibble  2.1.1       v dplyr   0.8.0.1
## v tidyr   0.8.3       v stringr 1.4.0
## v readr   1.3.1       v forcats 0.4.0
```

```
## -- Conflicts ------------------------------------------------------------------
----------- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
#data manipulation making ussample of 100000
datanew <- data %>%
  select(srch_destination_id, user_location_latitude, user_location_longitude, user_loca
tion_country, user_location_region, user_location_city)

destnew <- dest %>%
  select(srch_destination_latitude, srch_destination_longitude, srch_destination_id, src
h_destination_name)

new <- full_join(datanew, destnew, by = "srch_destination_id")

#chaning the state names to original and the column name to "region"
library(plyr)
```

```
## ------------------------------------------------------------------------
```

```
## You have loaded plyr after dplyr - this is likely to cause problems.
## If you need functions from both plyr and dplyr, please load plyr first, then dplyr:
## library(plyr); library(dplyr)
```

```
## ------------------------------------------------------------------------
```

```
##
## Attaching package: 'plyr'
```

```
## The following objects are masked from 'package:dplyr':
##
##     arrange, count, desc, failwith, id, mutate, rename, summarise,
##     summarize
```

```
## The following object is masked from 'package:purrr':
##
##     compact
```

```
new$user_location_region <- revalue(new$user_location_region, c("AL"="alabama","AK"="ala
ska","AZ"="arizona","AR"="arkansas","CA"="california","CO"="colorado","CT"="connecticut"
,"DC"="district of coulumbia","DE"="delaware","FL"="florida","GA"="georgia","HI"="hawai
i","ID"="idaho","IL"="illinois","IN"="indiana","IA"="iowa","KS"="kansas","KY"="kentucky"
,"LA"="louisiana","ME"="maine","MD"="maryland","MA"="massachusetts","MI"="michigan","MN"
="minnesota","MS"="mississippi","MO"="missouri","MT"="montana","NE"="nebraska", "NV"="ne
vada","NH"="new hampshire","NJ"="new jersey","NM"="new mexico","NY"="new york","NC"="nor
th carolina","ND"="north dakota","OH"="ohio","OK"="oklahoma","OR"="oregon","PA"="pennsyl
vania","RI"="rhode island","SC"="south carolina","SD"="south dakota","TN"="tennessee","T
X"="texas","UT"="utah","VT"="vermont","VA"="virginia","WA"="washington","WV"="west virgi
nia","WI"="wisconsin","WY"="wyoming"))

ussample<-new %>%
  filter (user_location_country == "UNITED STATES OF AMERICA")%>%
  sample_n(1000)
```

make csv

```
#dataset to csv
write.csv(ussample, file = "US_Sample.csv")


#attach csv
ussample <- read.csv("c:/Users/mkim458/Desktop/US_Sample.csv")
```

count user by state and change column and vector

```
detach("package:plyr", unload=TRUE)
```

```
## Warning: 'plyr' namespace cannot be unloaded:
##   namespace 'plyr' is imported by 'ggplot2' so cannot be unloaded
```

```
#counting users per region
usercount<- ussample  %>%
  select(user_location_latitude,user_location_longitude, user_location_region, user_loca
tion_city)%>%
  group_by(user_location_region)%>%
  mutate(Count = n())%>%
  filter(user_location_latitude != "NULL")


#change the column name
colnames(usercount)[colnames(usercount)=="user_location_region"]<-  "region"


#change to numeric
usercount$user_location_latitude <-as.numeric(usercount$user_location_latitude)
usercount$user_location_longitude <-as.numeric(usercount$user_location_longitude)
```

Draw US User Map

```
library(ggmap)
```

```
## Warning: package 'ggmap' was built under R version 3.5.3
```

```
## Google's Terms of Service: https://cloud.google.com/maps-platform/terms/.
```

```
## Please cite ggmap if you use it! See citation("ggmap") for details.
```

```
library(maps)
```

```
## Warning: package 'maps' was built under R version 3.5.3
```

```
##
## Attaching package: 'maps'
```

```
## The following object is masked from 'package:purrr':
##
##     map
```

```
#us map with state outline
us <- c(left = -125, bottom = 25.75, right = -67, top = 49)
map <- get_stamenmap(us, zoom = 5, maptype = "toner-lite")
```

```
## Source : http://tile.stamen.com/toner-lite/5/4/10.png
```

```
## Source : http://tile.stamen.com/toner-lite/5/5/10.png
```

```
## Source : http://tile.stamen.com/toner-lite/5/6/10.png
```

```
## Source : http://tile.stamen.com/toner-lite/5/7/10.png
```

```
## Source : http://tile.stamen.com/toner-lite/5/8/10.png
```

```
## Source : http://tile.stamen.com/toner-lite/5/9/10.png
```

```
## Source : http://tile.stamen.com/toner-lite/5/10/10.png
```

```
## Source : http://tile.stamen.com/toner-lite/5/4/11.png
```

```
## Source : http://tile.stamen.com/toner-lite/5/5/11.png
```

```
## Source : http://tile.stamen.com/toner-lite/5/6/11.png
```

```
## Source : http://tile.stamen.com/toner-lite/5/7/11.png
```

```
## Source : http://tile.stamen.com/toner-lite/5/8/11.png
```

```
## Source : http://tile.stamen.com/toner-lite/5/9/11.png
```

```
## Source : http://tile.stamen.com/toner-lite/5/10/11.png
```

```
## Source : http://tile.stamen.com/toner-lite/5/4/12.png
```

```
## Source : http://tile.stamen.com/toner-lite/5/5/12.png
```

```
## Source : http://tile.stamen.com/toner-lite/5/6/12.png
```

```
## Source : http://tile.stamen.com/toner-lite/5/7/12.png
```

```
## Source : http://tile.stamen.com/toner-lite/5/8/12.png
```

```
## Source : http://tile.stamen.com/toner-lite/5/9/12.png
```

```
## Source : http://tile.stamen.com/toner-lite/5/10/12.png
```

```
## Source : http://tile.stamen.com/toner-lite/5/4/13.png
```

```
## Source : http://tile.stamen.com/toner-lite/5/5/13.png
```

```
## Source : http://tile.stamen.com/toner-lite/5/6/13.png
```

```
## Source : http://tile.stamen.com/toner-lite/5/7/13.png
```

```
## Source : http://tile.stamen.com/toner-lite/5/8/13.png
```

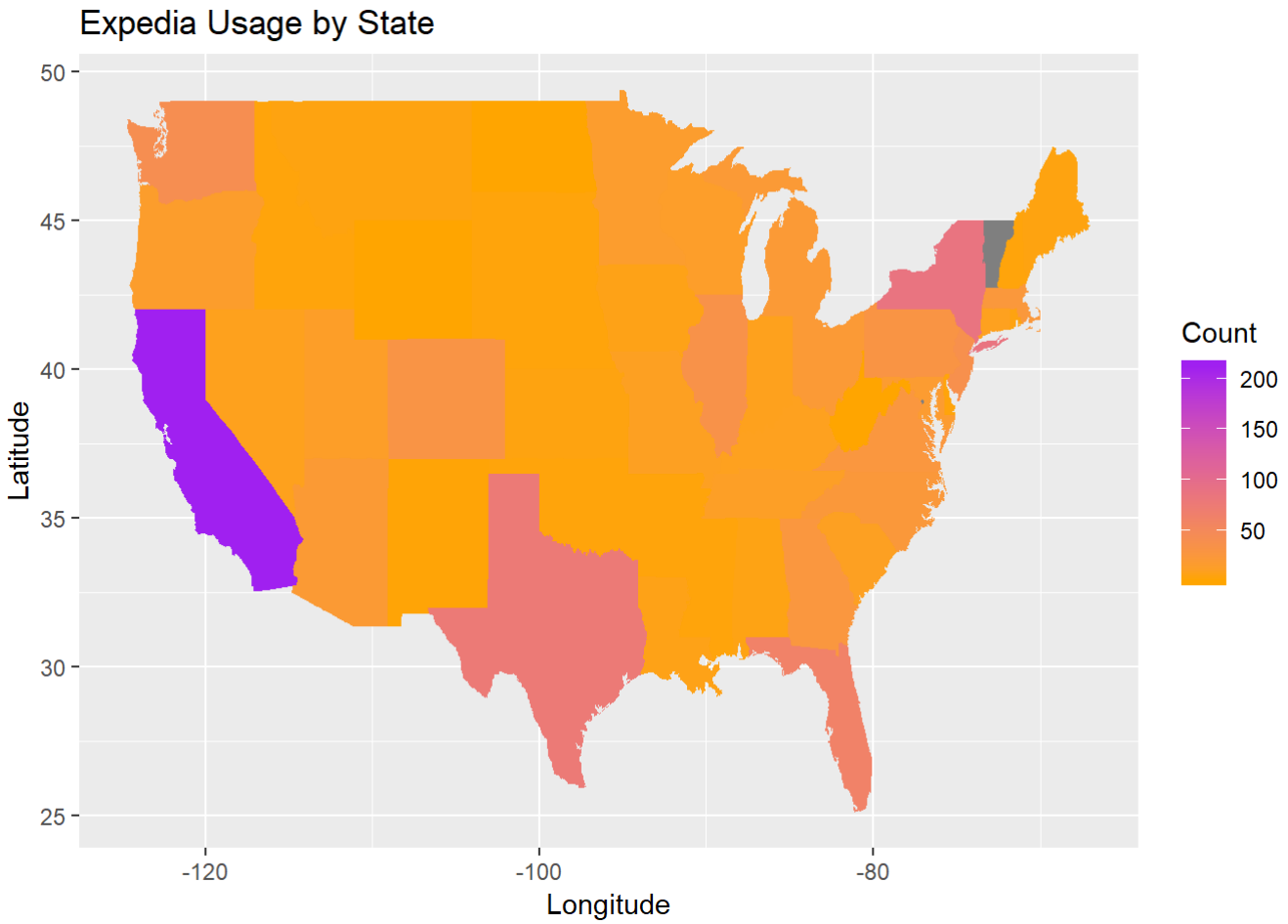
```
## Source : http://tile.stamen.com/toner-lite/5/9/13.png
```

```
## Source : http://tile.stamen.com/toner-lite/5/10/13.png
```

```
#joining state with usercount
states<-map_data("state")
usercountmap <- full_join(states, usercount, by = "region")
```

```
## Warning: Column `region` joining character vector and factor, coercing into
## character vector
```

```
#plot the map
qplot(long, lat, data=usercountmap, geom="polygon", group=group, fill=Count, main="Exped
ia Usage by State", xlab="Longitude", ylab="Latitude") + scale_fill_gradient(low = "oran
ge", high="purple")
```



separate country from srch_destination_name

```
#separate the city

ussample$srch_destination_name <-as.character(ussample$srch_destination_name)

dest <- list()
dest <- lapply(strsplit(ussample$srch_destination_name, ", "), rev)

for(i in seq_along(ussample$srch_destination_name)){
  ussample$srch_destination_name[i] <- dest[[i]][1]
}


#change column name
colnames(ussample)[colnames(ussample)=="srch_destination_name"]<-  "region"
```

counting the search destinations

```
#sampling
samp <- ussample%>% sample_n(1000)

#count the search destination
srchcount <- samp %>%
  group_by(region) %>%
  mutate(cnt=n())%>%
  select(cnt, srch_destination_latitude, srch_destination_longitude, region)

detach("package:ggmap", unload=TRUE)
detach("package:maps", unload=TRUE)
library(plyr)
```

```
## Warning: package 'plyr' was built under R version 3.5.3
```

```
## ------------------------------------------------------------------------------
```

```
## You have loaded plyr after dplyr - this is likely to cause problems.
## If you need functions from both plyr and dplyr, please load plyr first, then dplyr:
## library(plyr); library(dplyr)
```

```
## ------------------------------------------------------------------------------
```

```
##
## Attaching package: 'plyr'
```

```
## The following objects are masked from 'package:dplyr':
##
##     arrange, count, desc, failwith, id, mutate, rename, summarise,
##     summarize
```

```
## The following object is masked from 'package:purrr':
##
##     compact
```

```
srchcount$region <- revalue(srchcount$region, c("United States of America"="USA","United
Kingdom"="UK"))
```

Graph Search Destination in World Map

```
detach("package:plyr", unload=TRUE)
```

```
## Warning: 'plyr' namespace cannot be unloaded:
##   namespace 'plyr' is imported by 'ggplot2' so cannot be unloaded
```

```
library(ggmap)
```

```
## Warning: package 'ggmap' was built under R version 3.5.3
```

```
## Google's Terms of Service: https://cloud.google.com/maps-platform/terms/.
```

```
## Please cite ggmap if you use it! See citation("ggmap") for details.
```

```
library(maps)
```

```
## Warning: package 'maps' was built under R version 3.5.3
```

```
##
## Attaching package: 'maps'
```

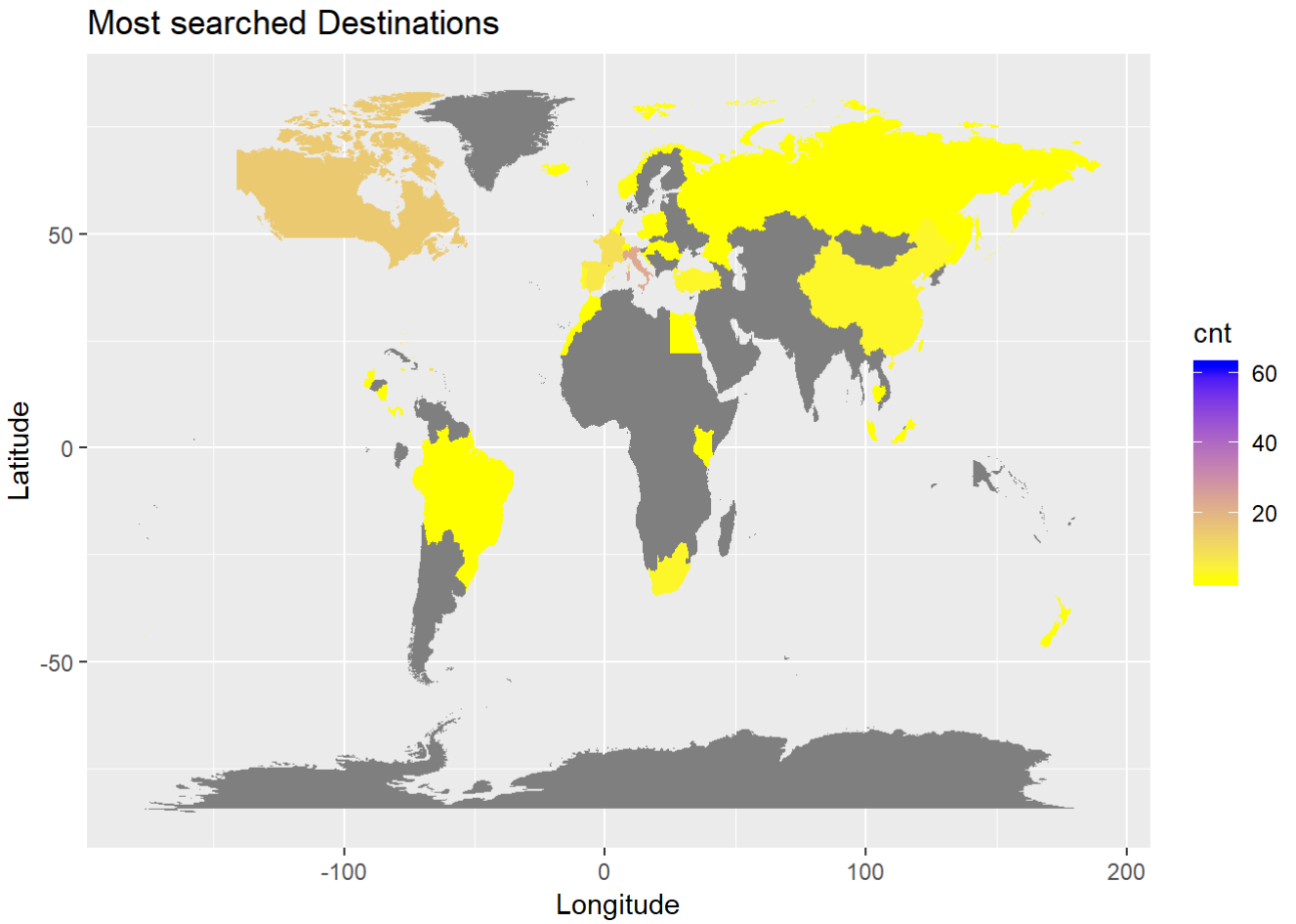
```
## The following object is masked from 'package:purrr':
##
##     map
```

```
#load world map
world <- map_data("world")

#join the world map with the count

final <- full_join(srchcount, world, by = "region")
final <- final %>% filter(region!= "USA")

#map the world with data
qplot(long, lat, data = final, geom = "polygon", group = group, fill=cnt, main="Most sea
rched Destinations", xlab="Longitude", ylab="Latitude") + scale_fill_gradient(low = "yel
low", high="blue")
```

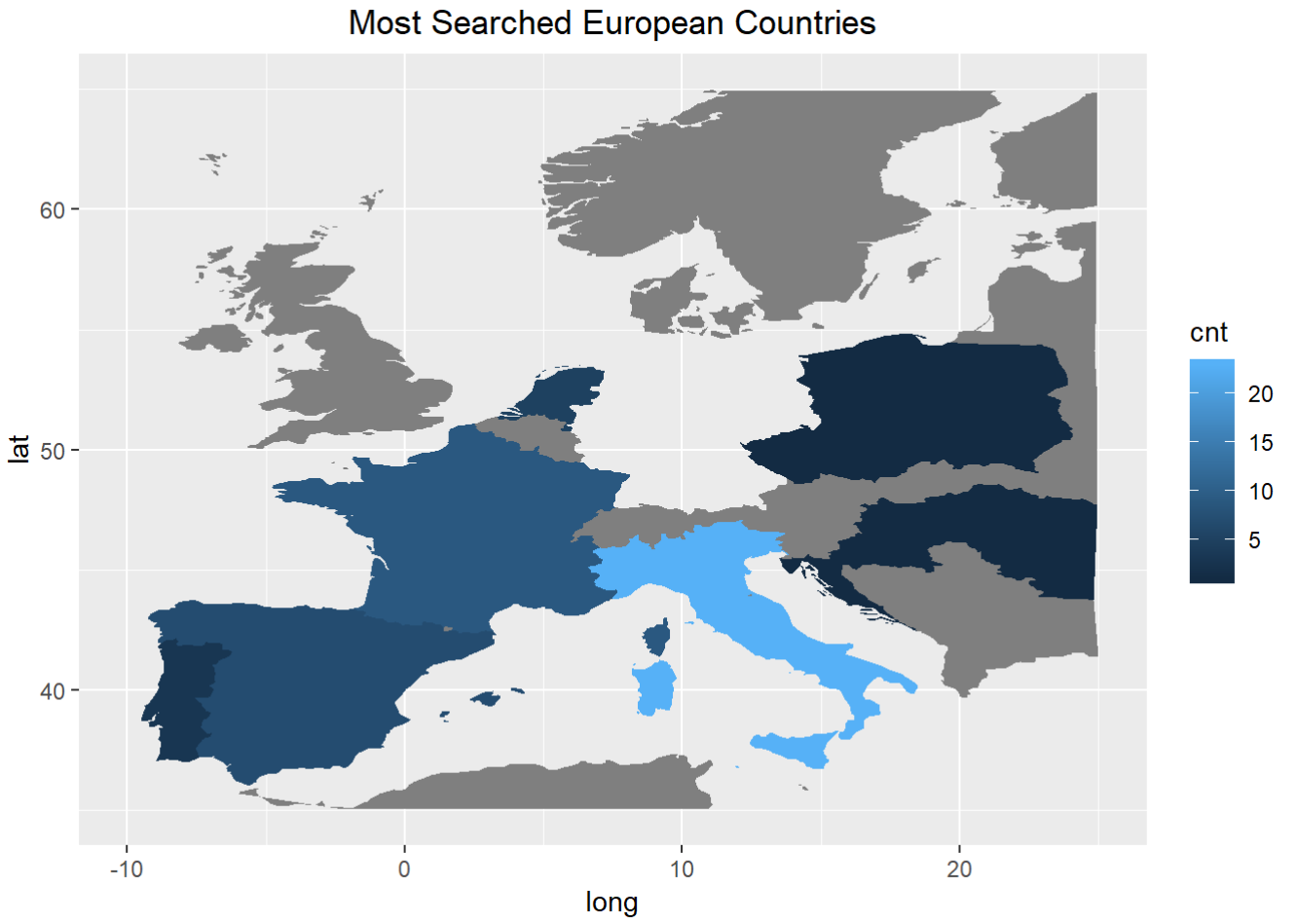


Graph the search only in Europe

```
library(ggmap)
library(maps)
```

OR

```
eumapf %>%
  filter(between(long, -10, 25),
         between(lat, 35, 65)) %>%
  ggplot(aes(x=long, y=lat, group=group)) +
  geom_polygon( aes(fill=cnt)) +
  ggtitle("Most Searched European Countries") +
  theme(plot.title = element_text(hjust = 0.5))
```



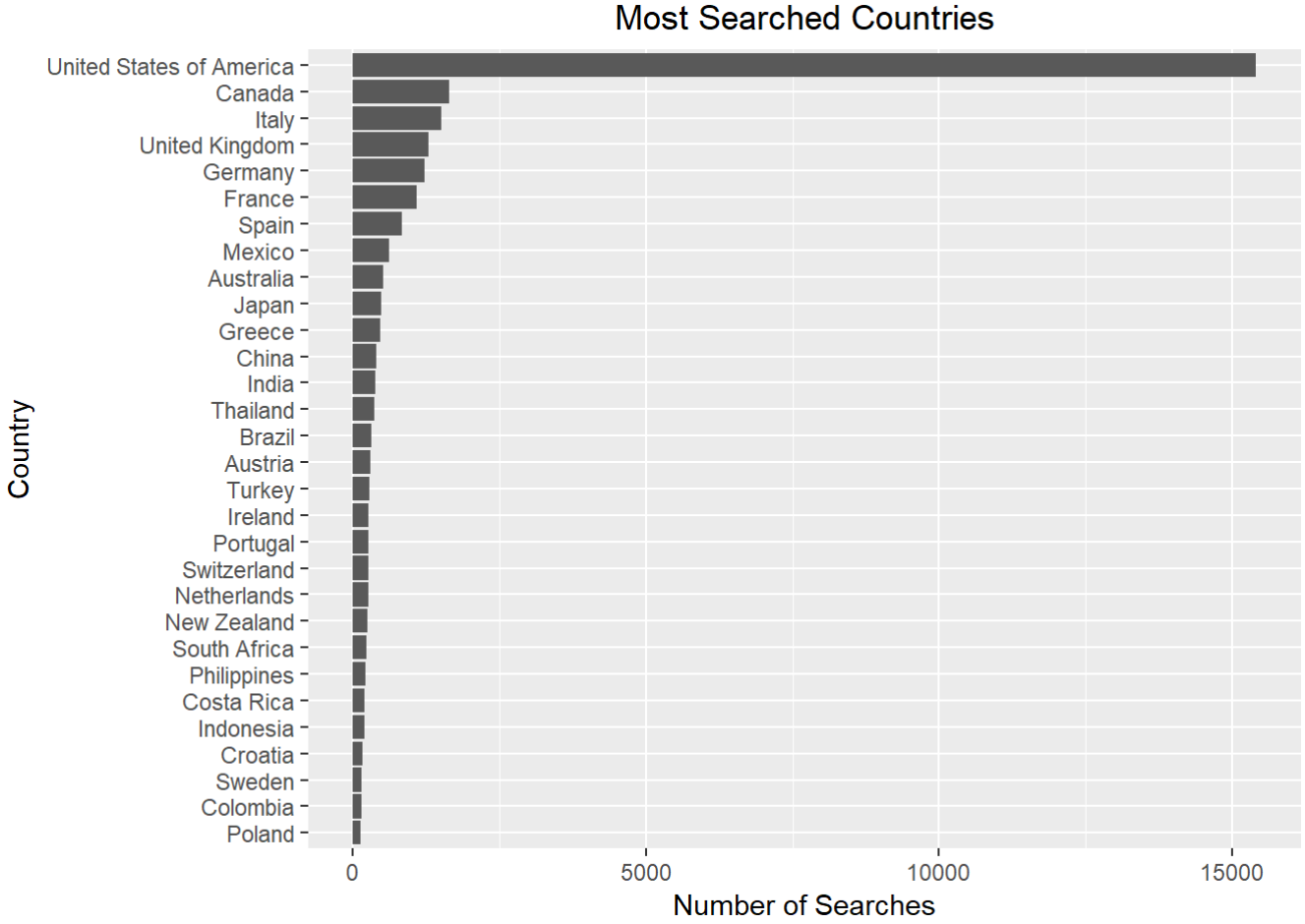
Graph by order which country was searched the most

```
sampledest <- destnew %>%
  select(srch_destination_name)

sampledest["n"] <- NA
sampledest$n <- 1

sampledest %>%
  separate(srch_destination_name, into = c("Street", "City", "region","country"), sep =
"," , fill = "left") %>%
  mutate(country=as.character(country)) %>%
  filter(country != "United States of America") %>%
  group_by(country) %>%
  summarise(sum = sum(n)) %>%
  top_n(30, sum) %>%
  ggplot(aes(x = fct_reorder(country, sum), y = sum)) +
    geom_col() +
    ggtitle("Most Searched Countries") +
  theme(plot.title = element_text(hjust = 0.5)) + coord_flip()+labs(y = "Number of Searc
hes", x = "Country")
```

```
## Warning: Expected 4 pieces. Additional pieces discarded in 9 rows [19605,
## 22014, 23227, 24185, 29420, 29616, 33101, 33906, 33981].
```



Looking at distance by family (adults and children) and only adults

```
distance<-data%>%
  select(orig_destination_distance, srch_adults_cnt, srch_children_cnt)%>%
  filter(orig_destination_distance != "NA")%>%
  filter(orig_destination_distance != "NULL")
```

```
distance$X <- NULL
```

create new column (binary: yes no) children and adult count together –> yes; if no children –> no

```
distance2 <- distance %>%
  mutate(dummy = ifelse(srch_children_cnt ==0, 0, 1))

distance3 <- distance2[sample(1:nrow(distance2), 1000, replace=F),]

distance3$dummy <- as.character(distance3$dummy)

distance3$orig_destination_distance <-as.numeric(distance3$orig_destination_distance)
```

if dummy is 1

```
library(plotly)
```

```
## Warning: package 'plotly' was built under R version 3.5.3
```

```
##
## Attaching package: 'plotly'
```

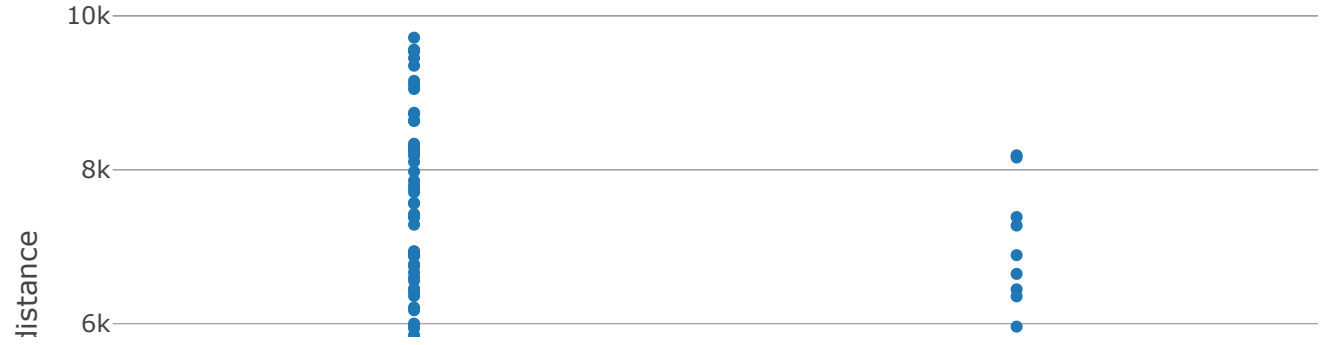
```
## The following object is masked from 'package:ggmap':
##
##     wind
```

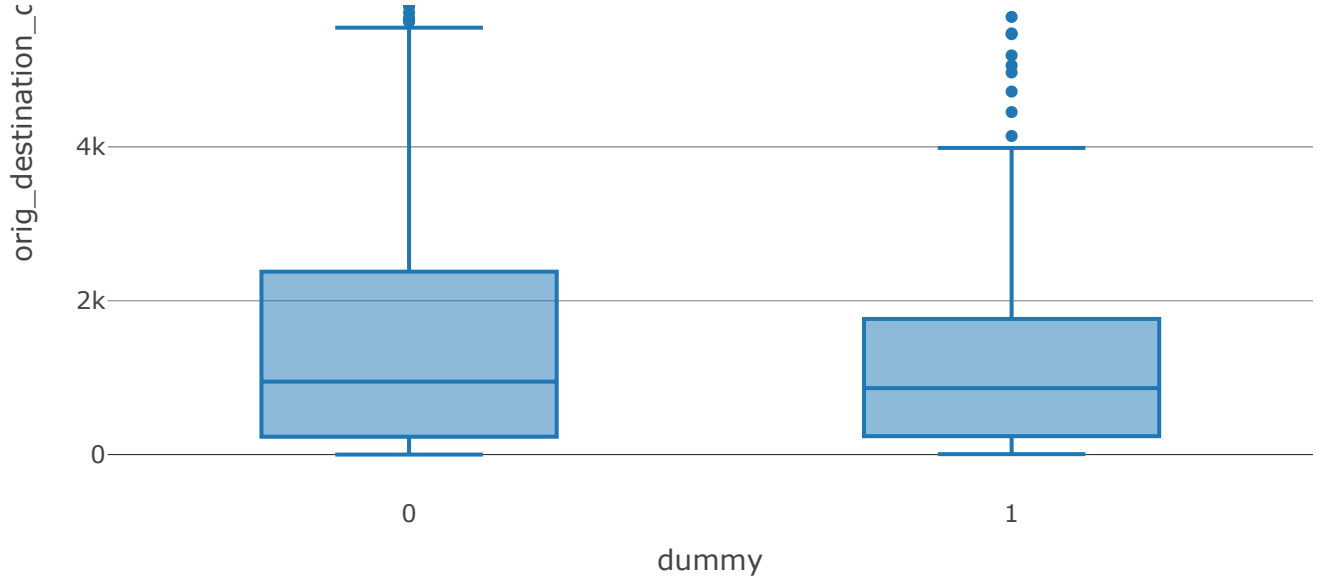
```
## The following object is masked from 'package:ggplot2':
##
##     last_plot
```

```
## The following object is masked from 'package:stats':
##
##     filter
```

```
## The following object is masked from 'package:graphics':
##
##     layout
```

```
plot_ly(distance3, x=~dummy, y=~orig_destination_distance, type="box")
```





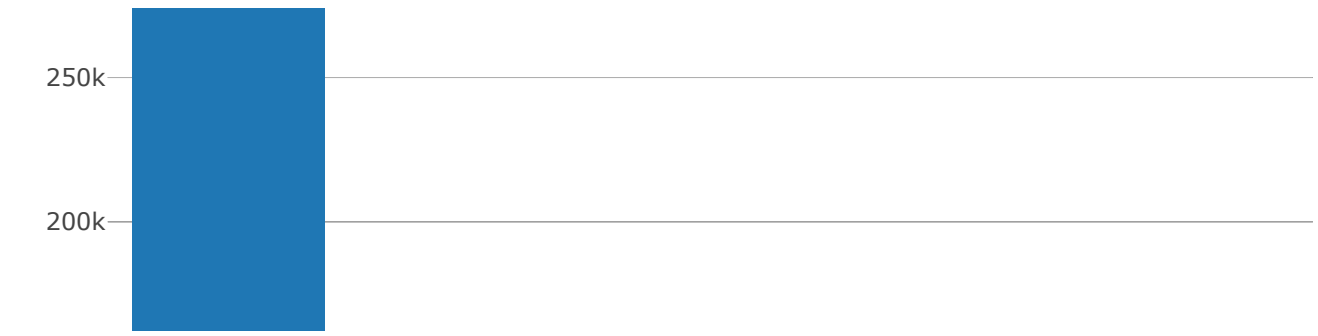
distance divided into (1) 0-1000 (2) 1000-2500 (3) 2500-5000 (4) 5000-7500 (5) >7500

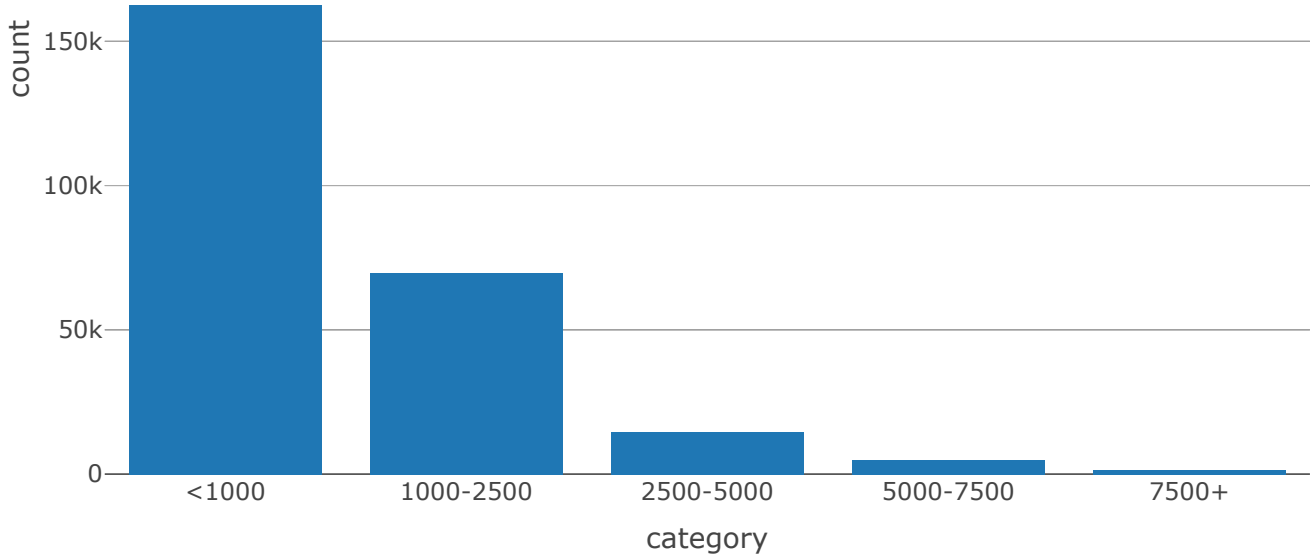
```
distance4 <- distance3 %>%
  group_by(orig_destination_distance) %>%
  mutate(category=orig_destination_distance)
```

```
distance4$category[distance4$orig_destination_distance < 1000] <- "<1000"
distance4$category[distance4$orig_destination_distance >= 1000 & distance4$orig_destinat
ion_distance <2500] <- "1000-2500"
distance4$category[distance4$orig_destination_distance >= 2500 & distance4$orig_destinat
ion_distance < 5000] <- "2500-5000"
distance4$category[distance4$orig_destination_distance >= 5000 & distance4$orig_destinat
ion_distance < 7500] <- "5000-7500"
distance4$category[distance4$orig_destination_distance >= 7500] <- "7500+"

distance5 <- distance4 %>%
  group_by(category) %>%
  mutate(count=n())


plot_ly(distance5, x = ~category, y =~count, type ="bar")
```





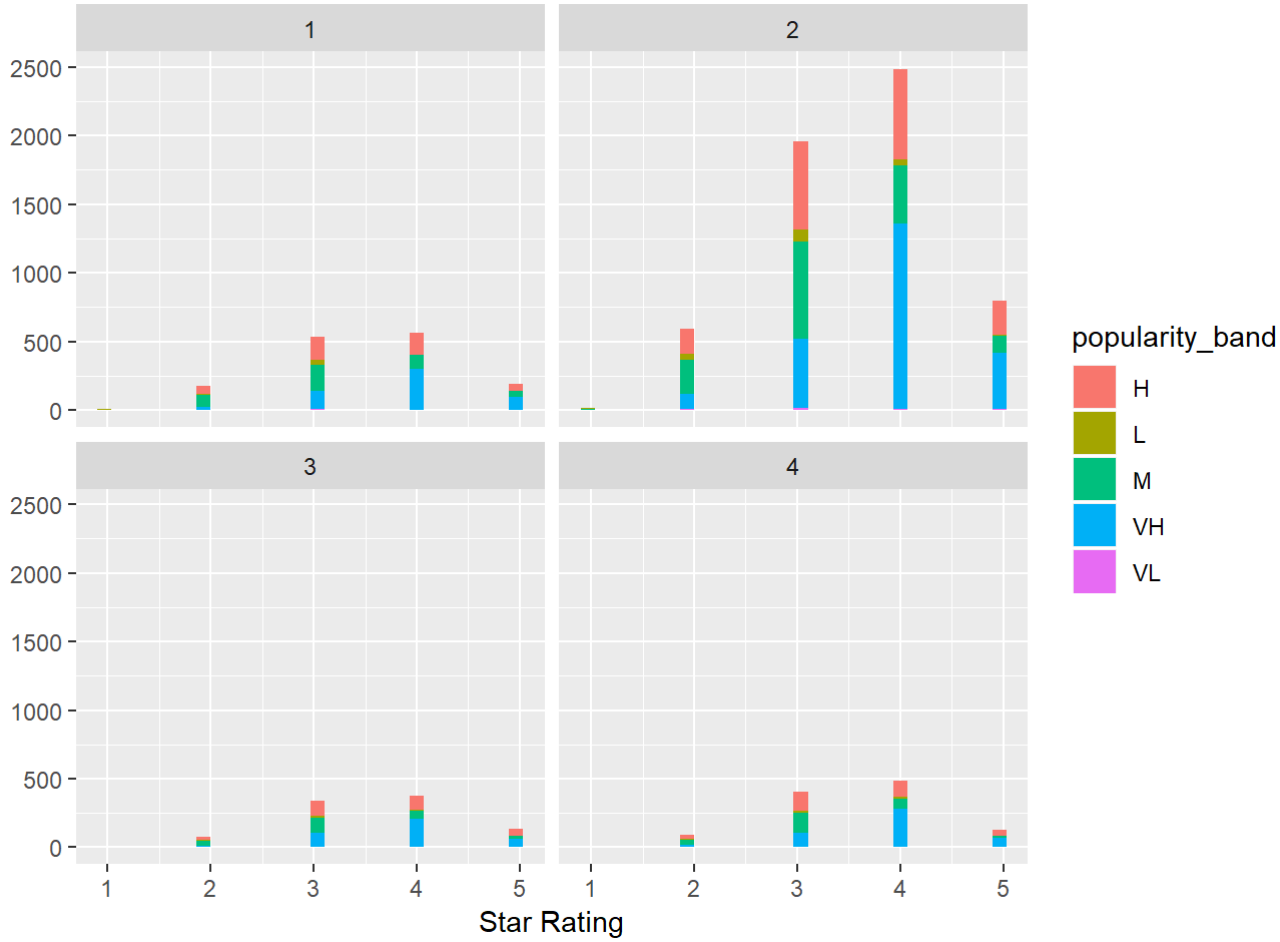
Hotel Star Rating

```
p1 <- data %>%
  sample_n(10000)

p2 <- p1 %>%
  select(prop_starrating, popularity_band, srch_adults_cnt, srch_children_cnt, is_bookin
g) %>%
  mutate(totaln = srch_adults_cnt + srch_children_cnt) %>%
  filter(totaln == 1 | totaln ==2 | totaln == 3 | totaln == 4) %>%
  filter(prop_starrating != 0)

qplot(prop_starrating, geom="histogram", facets=~totaln, fill = popularity_band, data=p
2) + labs(x= "Star Rating")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```


1
2
3
4
2500
2000
1500
1000
500
0
1
2
3
4
5
Star Rating
popularity_band
H
L
M
VH
VL